



Genetic Programming for Feature Selection and Feature Combination in Salient Object Detection

Shima Afzali^(✉), Harith Al-Sahaf, Bing Xue, Christopher Hollitt,
and Mengjie Zhang

School of Engineering and Computer Science,
Victoria University of Wellington, Wellington, New Zealand
{shima.afzali,harith.al-sahaf,bing.xue,
christopher.hollitt,mengjie.zhang}@ecs.vuw.ac.nz

Abstract. Salient Object Detection (SOD) aims to model human visual attention system to cope with the complex natural scene which contains various objects at different scales. Over the past two decades, a wide range of saliency features have been introduced in the SOD field, however feature selection has not been widely investigated for selecting informative, non-redundant, and complementary features from the existing features. In SOD, multi-level feature extraction and feature combination are two fundamental stages to compute the final saliency map. However, designing a good feature combination framework is a challenging task and requires domain-expert intervention. In this paper, we propose a genetic programming (GP) based method that is able to automatically select the complementary saliency features and generate mathematical function to combine those features. The performance of the proposed method is evaluated using four benchmark datasets and compared to nine state-of-the-art methods. The qualitative and quantitative results show that the proposed method significantly outperformed, or achieved comparable performance to, the competitor methods.

Keywords: Salient Object Detection · Genetic programming · Feature combination · Feature selection

1 Introduction

Human visual system can easily cope with the complex natural scene containing various objects at different scales using the visual attention mechanism. Salient Object Detection (SOD) aims to simulate the mentioned capability of the human visual system in prioritizing objects for high-level processing. SOD can be helpful to relieve complex vision problems such as scene understanding by detecting and segmenting salient objects [12]. Objects apparently catch more attention than background regions such as grass, sea, and sky. Therefore, if all generic objects can be detected in the first place, then scene understanding would be easily

performed at the subsequent stages. SOD serves as an important pre-processing step for many tasks, such as image classification, image retargeting and object recognition [12]. Several applications benefit from saliency detection such as: visual tracking, image and video compression, content-based image retrieval, human-robot interaction, object recognition [12].

SOD methods are broadly classified into two groups, bottom-up and top-down methods [29]. In the bottom-up methods, multiple low-level features, such as intensity, color, and texture, are extracted from an image to compute the saliency values. Top-down methods are task-dependent and they usually utilize domain specific knowledge [22]. SOD methods generally comprise the following steps [26]: (a) extract/design some saliency features from the input image, (b) compute individual feature maps using biologically plausible filters such as Gabor or difference of Gaussian filters, and (c) combine these different features to generate the final saliency map.

To date, domain experts have designed different types of saliency features such as local contrast, global contrast, edge density, backgroundness, focuses, objectness, convexity, spatial distribution, and sparseness [12]. Regarding the feature extraction stage, the design of new saliency features stimulates some possible questions. For example, whether the newly designed feature is informative for different image types (e.g., images with no much color variation, having cluttered background, having multiple objects, etc.), how does it effect other features, whether it is a duplicate feature (already exist) in the domain, which types of features it can complement, and how can we effectively use the new feature in different application scenarios. It is very difficult to provide definite answers for such questions. There are some potential ways to answer the aforementioned questions. One plausible way is to be a domain expert, this way suffers from some difficulties such as requiring domain knowledge of the task, domain experts are not always available and are very expensive to employ. Another way is developing a heuristic method which is very common in the literature [25,27]. However, it is becoming more and more challenging to design heuristic methods that are able to fully explore the potential of the existing features [12]. The next possible and suitable solution is to develop an automatic domain independent method. This method can widely explore characteristics of the existing and newly created features and find the relationship between them. In addition, it has the ability to select informative and non-redundant features that complement each other for different image types.

As mentioned before, the feature combination stage is one of the fundamental stages in SOD for generating the final saliency map [11]. In this regard, a few studies attempt to address the feature combination problem by finding the optimal values for the weights in the linear combination. For example, Liu et al. [23] employed the conditional random field (CRF) framework to learn the linear combination weights for different features. Afzali et al. [5] utilized particle swarm optimization (PSO) to learn a suitable weight vector for the saliency features and linearly combine the weighted features. Due to the highly nonlinearity of the visual attention mechanism, the above linear mapping might not perfectly

capture the characteristics of feature combination of human visual system. Consequently, nonlinear methods are required to fuse saliency features to achieve higher performance on different image types. Moreover, in the majority of existing methods [17, 22, 29], saliency features and the combination stage have been manually designed by domain experts. In this scenario, the feature extraction and combination tasks are highly dependent on domain-knowledge and human intervention.

Genetic programming (GP) which is a well-known evolutionary computation (EC) technique has the ability to tackle image-related problems, such as region detection, feature extraction, feature selection, and classification, in a wide variety of applications [7]. GP can automatically generates solutions, it is problem-independent, and it has a flexible representation and global search ability. The tree-based representation of GP makes it a suitable tool to do feature manipulation such as feature construction and feature selection. GP has a good capability in creating different solutions which is not thought about by domain experts. Moreover, GP is well-known for being flexible due to the ability of evolving various models, e.g., linear and non-linear models, and operating on different types of data such as numerical and categorical data [18]. The aforementioned properties of GP motivates us to utilize it for feature selection and feature combination problems in SOD.

1.1 Goal

This paper aims at utilizing GP to automatically select features from different level and scales, and combine those selected features for the task of saliency object detection.

- Introduce a new automatic GP-based feature selection and feature combination method;
- Formulate an appropriate fitness function to evaluate GP solutions (programs);
- Evaluate the proposed method using dataset of varying difficulties to test the generalisability property of this method; and
- Compare the performance of the proposed method to that of nine hand-crafted SOD methods to test whether those automatically evolved programs have the potential to achieve better or comparable performance to the domain-expert designed ones.

2 Background

2.1 Saliency Features

Jiang et al. [17] developed a discriminative regional feature integration (DRFI) approach which is a supervised SOD method. In DRFI, 93 features have been introduced, which contains three types of regional saliency descriptor including regional contrast, regional backgroundness, regional property. Regional contrast descriptor is a 29-dimensional feature vector contains color and texture features.

Color features are extracted from three different color spaces including RGB (red, green, and blue), HSV hue, saturation, and value), and $L^*a^*b^*$. For texture feature extraction, they used local binary patterns (LBP) feature [14] and responses of Leung-Malik (LM) filter bank [17]. Regional backgroundness descriptor that is a 29-dimensional feature vector extracted by computing the difference between each region and a pseudo-background region as a reference. Finally, regional property descriptor that is 35-dimensional feature vector computed by considering the generic properties of a region such as appearance and geometric features. Hence, a 93-dimensional ($2 \times 29 + 35$) feature vector is obtained.

2.2 Related Work

Over the past two decades [16], an extremely rich set of saliency detection methods have been developed. Most previous methods have focused on designing various hand-crafted features to find objects with visual features different than those from the background. As a pioneer, Itti et al. [16] proposed bottom-up visual saliency method using center-surround differences across multi-scale image features. Prior efforts employed simple features such as color and grayscale, edges, or texture, as well as more complex features such as objectness, focusness and backgroundness. The reader shall refer to the survey paper by Borji et al. [12] for more details on traditional saliency features.

In addition to the feature extraction, many SOD methods attempt to design different feature combination frameworks. Zhu et al. [30] used an optimization-based framework to combine multiple foreground and background features as well as the smoothness terms to automatically infer the optimal saliency values. Zhou et al. [29] developed a SOD method by integrating compactness and local contrast features. [29] addressed the drawback of combining global contrast and compactness features by considering local contrast, since local contrast can detect some salient regions ignored by compactness. In [29], a diffusion process based on manifold ranking is employed to propagate saliency information. Lin et al. [22] introduced a method to predict salient object by extracting multiple features such as local contrast, global contrast, and background contrast in different feature extraction scales, e.g., pixel-level, region-level and object-level. In [22], the authors manually designed a framework to integrate the features, e.g., background priors, refined global contrast, and local contrast. Liu et al. [23] employed the conditional random field to learn an optimal linear combination of local, regional, and global features. Jiang et al. [17] used random forest to the fusion weights of feature maps. In [17], they used three types of regional saliency features including contrast, backgroundness, and property to 93-dimensional feature vector for each region. The majority of the aforementioned studies suffers from requiring domain-knowledge and human intervention in designing a good feature combination method. Hence, recent studies attempt to address the mentioned problem by developing automatic approaches.

Compared with traditional methods that use hand-crafted features, convolutional neural network (CNN) based methods that adaptively extract high-level semantic information from raw images have shown impressive results in predicting saliency maps [28]. Lee et al. [20] considered both hand-crafted features

and high-level features extracted from CNNs. To combine the features together, they designed a unified fully connected neural network to compute saliency maps. Recently, Hou et al. [15] developed a CNN method which combines both low-level and high-level features from different scales. Although CNN based developments have achieved high performance in the SOD domain in recent years, the top CNN methods require nontrivial steps such as generating object proposals, applying post-processing, enforcing smoothness through the use of superpixels or defining complex network architectures [24].

Among EC techniques, GP has the ability to solve various complex problems in many research areas such as feature extraction, classification, and object detection [10]. Lensen et al. [21] developed a GP approach to automatically select regions, extract histogram of oriented gradients (HOG) features and perform binary classification on a given image. Al-Sahaf et al. [8] showed that GP has the ability to automatically extract features, perform feature selection and image classification. Later on, Al-Sahaf et al. [9], used multitree GP representation to automatically evolve image descriptors. Unlike existing hand-crafted image descriptors, [9] automatically extracts feature vectors. Afzali et al. [4] utilized GP to construct informative foreground and background features from the extracted saliency features, then combined the constructed two features employing a spatial blending phase. However, this method was not fully automatic and required human intervention in the feature combination stage. Ain et al. [6] developed GP-based method to do feature selection and feature construction for skin cancer image classification. The authors claimed that the GP selected and constructed features helped the classification algorithms to produce effective solutions for the real-world skin cancer detection problem. In the mentioned studies, GP made the proposed approaches free from any requirement for human intervention or domain-specific knowledge. Apart from the existing GP methods, this study investigates using GP for feature selection and feature combination in SOD field.

3 The Proposed Method

This section describes the proposed GP based method for automatically feature selection and feature combination (GPFSFC) for saliency detection. The overall structure is depicted in Fig. 1. For the training stage, first, different image segmentation-levels are computed for each image in the training set, then saliency features are extracted from the segmented images. Second, the saliency feature set and ground truth are fed into GP. Third, the GP process generates and evaluates the GP programs. Finally, the GP process results 50 evolved individuals. For the validation stage, after completing the segmentation and feature extraction parts, the saliency feature set and ground truth of the validation images are used to select the best individual from the evolved GP individuals. For the test stage, for a given test image, similar to the training and validation stages, multi-level image segmentation and feature extraction are computed. Then, the saliency map of the image is produced by employing the selected GP individual.

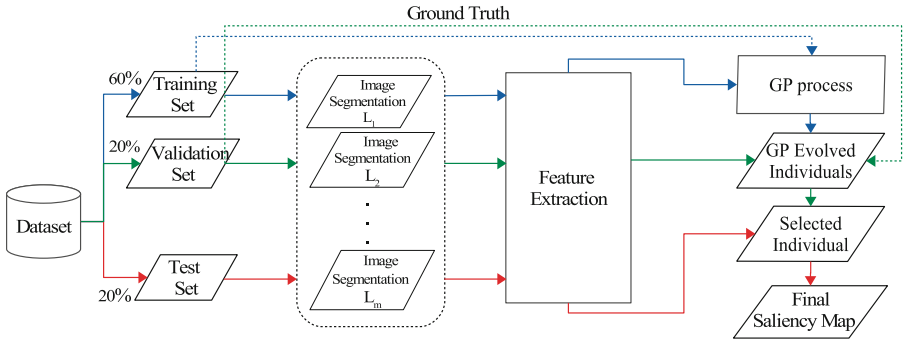


Fig. 1. The overall algorithm of GPFSC.

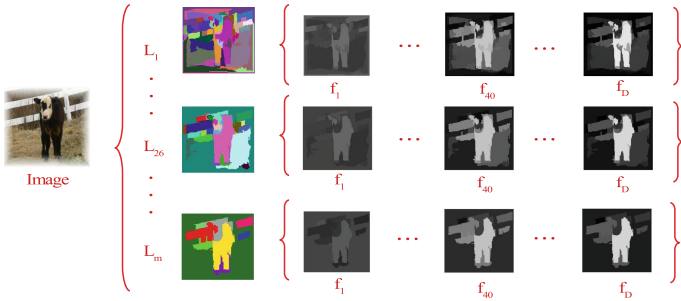


Fig. 2. Feature extraction from different segmentation levels.

3.1 Feature Selection and Feature Combination by GP

GP evolves the initial population using the ramped-half-and-half method. Each individual in the population takes saliency features and constants as terminals and combines them using the operations from the function set. The goodness of each individual is evaluated by a fitness function (more details below). The fitness value of each individual is computed by taking the average of the performance over all training images. In the subsequent generation, a population of new individuals are generated by applying the different genetic operators such as crossover, mutation and elitism on some individuals selected from the current population. This process continues until the maximum number of generations is reached, and the best evolved program is then returned which is a mathematical function of different selected operations and features. The terminal set, function set and fitness function are presented and discussed in detail in the following sections.

Terminal Set. In order to provide the terminal set for the GP process, the following preprocessing steps are employed. Firstly, for a given image I , a set of m -level segmentations $L = \{L_1, L_2, \dots, L_m\}$ is computed, each segmentation is a decomposition of the image I (Fig. 2). Here, the graph-based image segmentation method [13] is employed to generate multiple segmentations using m groups of different parameters. In this study, m is set to 48 by following [17]. Second, each region of a segmentation level is represented by D -dimensional feature vector (Fig. 2). D is set to 103 (10 + 93) by collecting 10 saliency features from [4] and 93 features from the DRFI method [17]. Since a large number of saliency features have been introduced in the literature, it is worthwhile to develop an automatic method which can explore and tackle with the large search space of different features. In this study, we provided a wide range of features to investigate how GP can cope with. Figure 2 demonstrates visualizations of different segmentation levels and saliency features belong to each level.

Function Set. The function set is made up from three arithmetic operators, one trigonometric function and one conditional function, which are $\{+, -, \times, \sin, if\}$. The first three arithmetic operators and the trigonometric operator have their regular meaning, and if operator takes three input arguments and returns the second argument if the first is less than the second; otherwise, it returns the third argument.

Fitness Function. The proposed fitness function is based on the Kullback-Leibler (KL) divergence (also called relative entropy) [19], which measures the difference between two probability distributions. The fitness value is computed by taking the average of the KL value over all training images as follow

$$Fitness = \frac{1}{n} \sum_{i=1}^n KL(I_i, G_i) \quad (1)$$

where, n is the number of images, I_i and G_i are the i^{th} image (the output saliency image computed using GP) and its corresponding ground truth. We utilize the KL divergence to see whether the output of GP is closer to the ground truth. Before applying KL divergence, we apply softmax on the output of GP and the ground truth to compute the probability of each one for the KL divergence. The KL divergence is defined as

$$KL(p, q) = \sum_{r \in R} p(r) \frac{\ln p(r)}{q(r)} \quad (2)$$

where r is a region from the region vector R . p and q are two probability distributions of the ground truth and the output of GP, respectively.

4 Experiment Design

4.1 Benchmark Datasets

In this paper, we evaluate the performance of the proposed method using four benchmark datasets in SOD. We choose these datasets based on the following four criteria: (1) being widely-used, (2) containing both large and small number of images, and (3) having different biases (e.g. number of salient objects, image clutter, center-bias). We split each dataset into three parts: a training set (60%), a validation set (20%) and a test set (20%).

The first dataset is single-object (*SED1*), which is a subset of the SED dataset [12]. The SED1 dataset includes 100 images containing only one salient object in each image. Pixel-wise ground truth annotations for the salient objects in SED1 are provided. Here, we employed the SED1 dataset, since we only considered single salient object images.

The second dataset is ASD, which is a subset of the MSRA10K dataset [23]. The MSRA10K dataset provides bounding boxes manually drawn around salient regions by nine users. However, a bounding box-based ground truth is far from being accurate. Thus, [23] created an accurate object-contour based ground truth dataset of 1000 images. Each image is manually segmented into foreground and background. Most images have only one salient object and strong contrast between objects and backgrounds.

The third dataset is ECSSD dataset which contains 1000 semantically meaningful but structurally complex images [12]. In contrast to some simple datasets such as MSRA, in which background structures are simple and smooth, the ECSSD dataset contains more complex image types. Ground truth masks are provided by 5 subjects.

The fourth dataset is PASCAL which contains 850 images [11]. The reason to use PASCAL-S datasets was to assess performance of the proposed method over scenes with multiple objects with high background clutter.

In this study, the raw images and respective ground truth are re-sized to 200×200 pixels to leverage computational efficiency.

Following [12], we utilized three universally-agreed, standard, and easy-to-compute evaluation criteria, which are precision-recall (PR) curve, receiver operating characteristic (ROC) curve and F-measure to evaluate the different SOD methods. We used these criteria to compute the quantitative performance of each SOD method.

The PR curve is obtained by binarizing the saliency map using a number of thresholds ranging from 0 to 255, as in [25]. To compare the quality of the different saliency maps, the threshold is varied from 0 to 255, and the precision and recall at each value of the threshold are computed by comparing the binary mask and the corresponding ground truth. Then a precision-recall curve is plotted using the sequence of precision-recall pairs. Precision corresponds to

the fraction of the pixels correctly labeled against the total number of pixels assigned salient, whereas recall is the fraction of the pixels correctly labeled in relation to the number of ground truth salient pixels.

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

where TP (true positive) is the number of foreground pixels that are correctly detected as foreground, FP (false positive) is the number of background pixels that are incorrectly detected as foreground, and FN (false negative) is the number of foreground pixels that are incorrectly detected as background.

The ROC curve can also be generated based on the true positive rates (TPR) and false positive rates (FPR) obtained during the calculation of the PR curve. TPR and FPR can be computed as

$$TPR = \frac{TP}{TP + FN} \quad (5)$$

$$FPR = \frac{FP}{FP + TN} \quad (6)$$

An image dependent adaptive threshold T_a proposed by Achanta et al. [1] is used to binarize the saliency map (\mathbf{S}). T_a is computed as twice as the mean saliency of \mathbf{S} .

$$T_a = \frac{2}{W \times H} \sum_{x=1}^W \sum_{y=1}^H \mathbf{S}(x, y) \quad (7)$$

where W and H are the width and height of the saliency map \mathbf{S} , respectively, and $\mathbf{S}(x, y)$ is the saliency value of the pixel at position (x, y) .

Often, neither precision nor recall can fully evaluate the quality of a saliency map. To this end, the F-measure (F_β) (Eq. 8) is used as the weighted harmonic mean of precision and recall with a non-negative weight β^2 .

$$F_\beta = \frac{(1 + \beta^2)Precision \times Recall}{\beta^2 Precision + Recall} \quad (8)$$

As suggested in many salient object detection works [1, 25], we set β^2 to 0.3, to weight precision more. The reason is because recall rate is not as important as precision [23]. For instance, 100% recall can be easily achieved by setting the whole map to be foreground.

4.2 Parameter Tuning

GP has a number of parameters which can be altered for a given problem. In this study, the initial population is created by ramped half-and-half method. The population size is restricted to 300 individuals, since the computational costs of dealing with images is high. The minimum tree depth is 2 and the maximum

Table 1. GP parameters.

Parameter	Value	Parameter	Value
Population size	300	Initial population	Half-and-half
Generations	50	Tree depth	2–10
Mutation rate	0.40	Selection type	Tournament
Crossover rate	0.60	Tournament size	7

depth is 10. The evolutionary process is terminated when the maximum number of 50 generations is reached. The evolutionary process is independently executed 30 times using different random seed values and the average performance is reported. Here, the mutation and crossover rates are set to 40% and 60%, respectively, based on the fact that a higher mutation rate could produce better training performance by allowing a wider exploration of the search space. The best evolved program is kept to prevent the performance of the subsequent generation from degrading. The tournament selection method is used for selecting individuals for the mating process and the tournament size is set to 7. Table 1 gives a summary for the GP parameters.

4.3 Methods for Comparison

The proposed method is compared to nine state-of-the-art methods, five methods including DRFI, GS, GMR, SF, and RBD are selected from [12], and four other methods including MSSS [2], wPSO [5], GPFBC [4], and FBC [3].

5 Results and Discussions

5.1 Quantitative Comparisons

In Fig. 3(a) and (b), precision-recall and ROC curves of GPFSFC is comparable with DRFI and outperforms other methods on the SED dataset. Figure 3(c) shows that GPFSFC has similar average precision and F-measure results to DRFI, but higher average recall than DRFI. On the SED dataset, DRFI and GPFSFC have generally good results regarding the average precision, recall, and F-measure among the other SOD methods.

As it can be seen in Fig. 4(a) and (b), GPFSFC is performing as the second best method after DRFI regarding the precision-recall and ROC curves on the ASD dataset. In Fig. 4(c), although GPFSFC has slightly lower average recall than DRFI, RBD, GS, and FBC on the ASD dataset, GPFSFC has the highest average precision and F-measure among all the other methods. Regarding precision-recall and ROC curves in Fig. 5(a) and (b), GPFSFC is the third good performing method among the nine state-of-the-art methods, where DCNN outperforming all the methods, although it could not perform as good as GPFSFC and DRFI on the SED and ASD datasets. In Fig. 6 similar to Fig. 5, GPFSFC shows the good results after DCNN and DRFI on the PASCAL dataset.

Considering quantitative results, although GPFSFC loses its performance to DCNN on the ECSSD and PASCAL datasets, it shows good performance comparing to DCNN on the SED and ASD datasets. This limitation is mainly due to lack of high-level features in the feature set of GPFSFC, while DCNN benefits of combining both low-level and high-level features. Extracting more powerful feature representations and training the method with complex scenes can be used to deal with the challenging datasets.

To compare GPFSFC with DRFI, as DRFI employed an ensemble learning containing a large number of decision trees to predict the saliency value of the regions, it can generally generate more accurate results than the automatically evolved GP programs which is only one independent tree in this experiment.

5.2 Qualitative Comparisons

The qualitative comparisons of GPFSFC and nine other state-of-the-art methods are illustrated in Figs. 7 and 8. For the qualitative comparison, multiple representative images are selected from different datasets which incorporate a variety of difficult circumstances, including complex scenes, salient objects with center bias, salient objects with different sizes, low contrast between foreground and background. Figure 7 presents some challenging cases where GPFSFC can successfully highlight the salient object and suppress the background. For example, the first row is a complex image where the building is not homogeneous, it has reflection on the water and complex background. However, GPFSFC and DRFI can deal with it, while other methods such as DCNN, RBD, GMR, GS, SF, and MSSS performs poorly to detect the object. DRFI is good in detecting object, but it wrongly highlights some part of the background regions.

In 4th row, GPFSFC can completely suppress background where the background is cluttered. This is due to the advantage of selecting informative background features by the evolved GP program. As can be seen in the 5th image, it has non-homogeneous foreground and complex background, GPFSFC shows good performance, while the other nine methods are struggling in both highlighting the object and completely suppressing the background. In the 6th row, GPFSFC properly covers the foreground object, although the color contrast between the object and background is low. Choosing informative contrast, backgroundness, and property (appearance and geometric) features and combining them using suitable mathematical operations is the key point for having good performance on the aforementioned images.

Figure 7 shows that DCNN fails to completely detect salient object when the image has complex background and low contrast between the foreground and background. One potential reason can be lack of segment-level information like prior knowledge on segment level.

Although both GPFSFC and DRFI show good performance in different scenarios, these methods suffer in some challenging cases such as images in Fig. 8. GPFSFC fails to completely identify the foreground object in all three images and wrongly highlights the background in the 2nd and 3rd images. This problem is due to the lack of the high-level knowledge and enough training samples to learn different scenarios and object types.

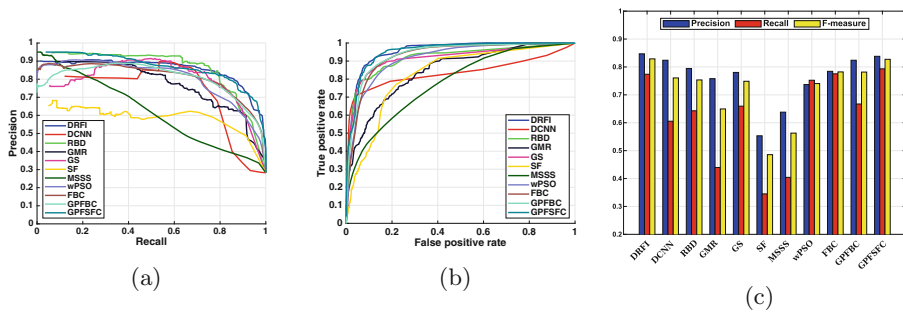


Fig. 3. Performance of GPFSC compared to nine other methods on SED1.

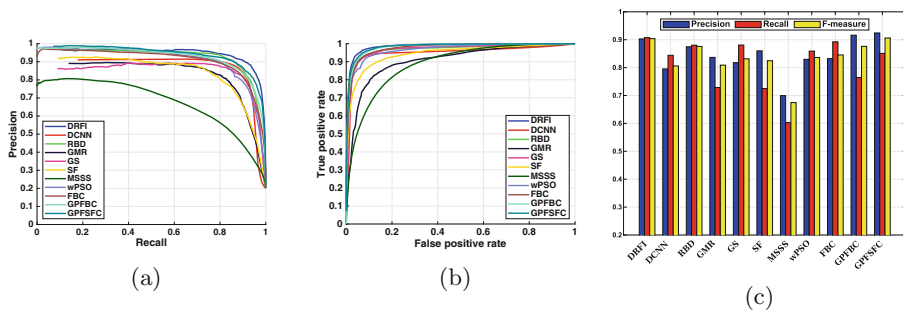


Fig. 4. Performance of GPFSC compared to nine other methods on ASD.

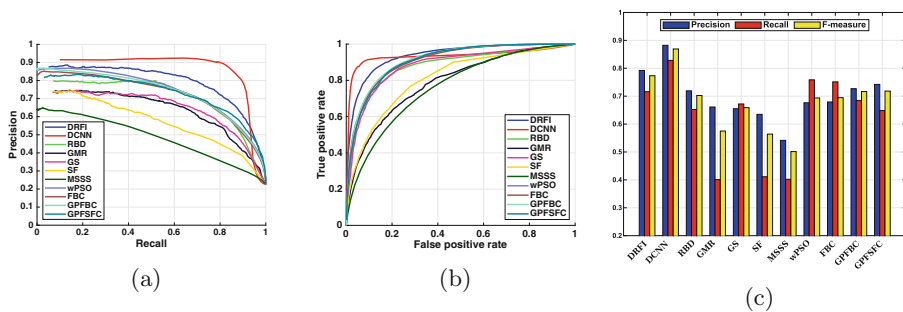


Fig. 5. Performance of GPFSC compared to nine other methods on ECSSD.

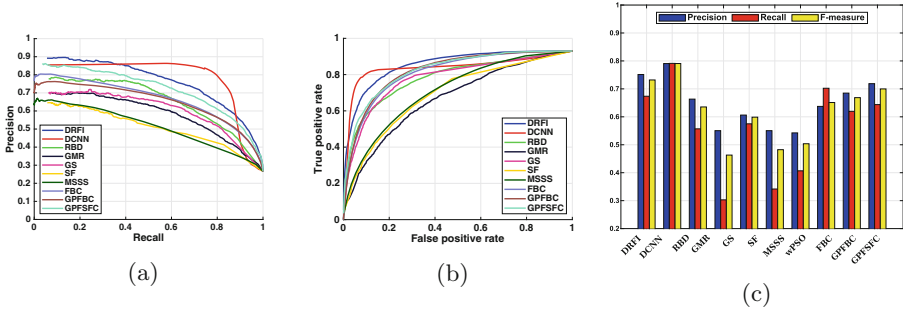


Fig. 6. Performance of GPFsFC compared to nine other methods on PASCAL.

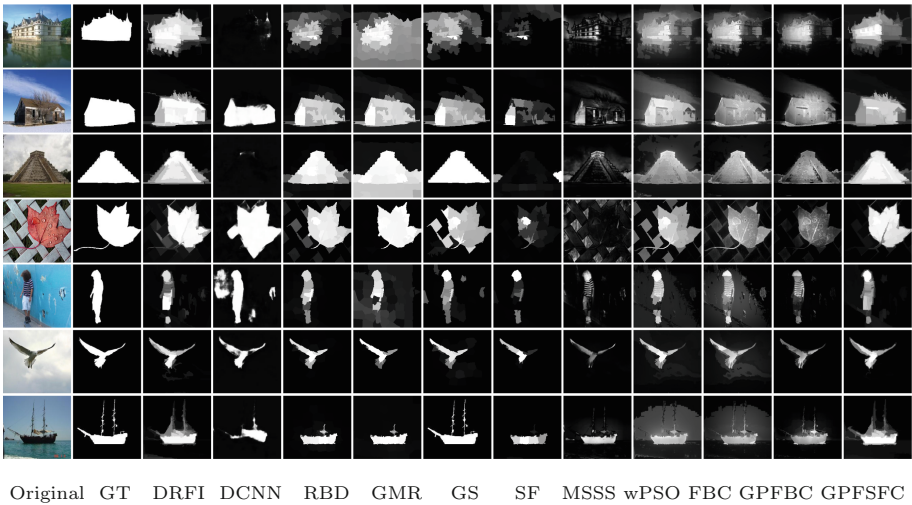
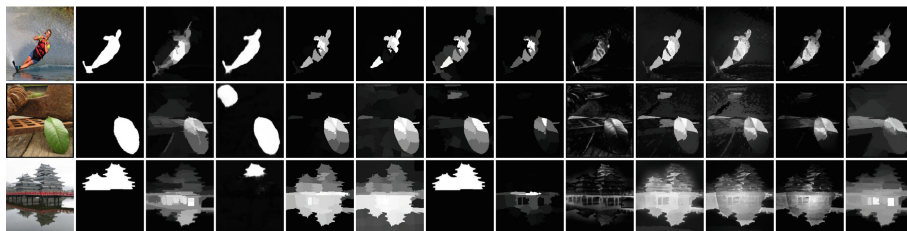


Fig. 7. Some visual examples of tGPFsFC and nine other SOD methods.

5.3 Further Analysis

Figure 9 shows an example of evolved GP program with high performance on the ASD dataset. Overall, there are 27 nodes in this program where 14 nodes are leaves and the other 13 are functions. The description of the selected features by GP represented in Table 2. As it can be seen in Fig. 9, five regional backgroundness features $\{f_{32}, f_{34}, f_{35}, f_{36}, f_{40}\}$ are selected to suppress background regions. Three regional property features $\{f_{65}, f_{66}, f_{71}\}$ are selected to consider the generic properties of regions. Finally, three contrast features $\{f_0, f_{94}, f_{96}\}$ are chosen to capture the color differences space (changes), as a region is likely thought to be salient if it is different from the other regions. This GP program only chooses 11 features from 103 features and decrease the dimensionality nearly nine times. The GP process considers complementary characteristic of



Original GT DRFI DCNN RBD GMR GS SF MSSS wPSO FBC GPFBC GPFSC

Fig. 8. Some visual examples of GPFSC and nine other SOD methods.

Table 2. Selected feature’s description.

Feature	Description	Feature type
f_0	Average R value [17]	Regional contrast
f_{32}	Average B value [17]	Regional backgroundness
f_{34}	Average H value [17]	Regional backgroundness
f_{35}	Average S value [17]	Regional backgroundness
f_{36}	Average V value [17]	Regional backgroundness
f_{40}	Average b^* value [17]	Regional backgroundness
f_{65}	Normalized perimeter [17]	Regional property
f_{66}	Aspect ratio of the bounding box [17]	Regional property
f_{71}	Variiances of the a^* value [17]	Regional property
f_{94}	Local contrast [23]	
f_{96}	Global contrast [23]	

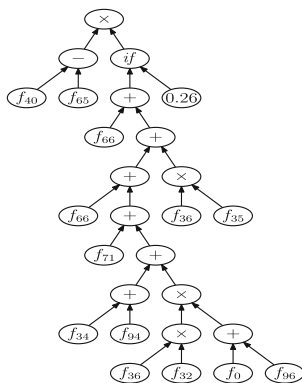


Fig. 9. Sample program evolved by GP.

features in feature selection and combination stages using the fitness value of the evolved GP programs. Figure 9 demonstrates that the evolved GP program or mathematical expression is a non-linear function. Furthermore, it represents a good example to present how the combination of different types of features such as color, backgroundness, appearance and geometric is important in properly detecting salient object.

6 Conclusions

In this study, GP has been successfully employed to automatically select and combine saliency features to produce the final saliency map. The proposed method can easily incorporate any additional features and select the features complement each other. It makes no assumption of linear superposition or equal weights of features and it does not require domain-expert. GPFSC has the ability to tackle a wide range of saliency features from different segmentation levels and explore various mathematical expressions for the feature combination stage. The saliency features by themselves are not that accurate and sufficient to properly detect the salient object and suppress background. Therefore, a good feature selection and combination method plays an important role in achieving high performance. The quantitative and qualitative results reveal that GPFSC can effectively choose the features which complement each other and have good effect on each other, thus, the final combination of those features results in a good saliency map. In this paper, although GPFSC was slightly worse on the ECSSD and PASCAL datasets, it showed promising results by outperforming one of the well-know and recent CNN methods (DCNN) on two datasets, i.e., SED and ASD.

For future work, as GP showed promising results for feature selection, feature construction, and feature combination processes, it is worth studying the use of GP for automatically extracting saliency features from the row images and investigate whether GP is successful in generating new features that are as informative as the hand-crafted features. Moreover, we would like to explore GP for extracting semantic (high-level) saliency features and provide further analysis on the extracted features.

References

1. Achanta, R., Hemami, S., Estrada, F., Susstrunk, S.: Frequency-tuned salient region detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1597–1604. IEEE (2009)
2. Achanta, R., Süsstrunk, S.: Saliency detection using maximum symmetric surround. In: Proceedings of the 17th IEEE International Conference on Image Processing, pp. 2653–2656. IEEE (2010)
3. Afzali, S., Al-Sahaf, H., Xue, B., Hollitt, C., Zhang, M.: Foreground and background feature fusion using a convex hull based center prior for salient object detection. In: Proceedings of the 33rd International Conference on Image and Vision Computing New Zealand, pp. 1–6. Springer (2018)

4. Afzali, S., Al-Sahaf, H., Xue, B., Hollitt, C., Zhang, M.: A genetic programming approach for constructing foreground and background saliency features for salient object detection. In: Mitrovic, T., Xue, B., Li, X. (eds.) *AI 2018. LNCS (LNAI)*, vol. 11320, pp. 209–215. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03991-2_21
5. Afzali, S., Xue, B., Al-Sahaf, H., Zhang, M.: A supervised feature weighting method for salient object detection using particle swarm optimization. In: *Proceedings of the IEEE Symposium Series on Computational Intelligence*, pp. 1–8. IEEE (2017)
6. Ain, Q.U., Xue, B., Al-Sahaf, H., Zhang, M.: Genetic programming for feature selection and feature construction in skin cancer image classification. In: Geng, X., Kang, B.-H. (eds.) *PRICAI 2018, Part I. LNCS (LNAI)*, vol. 11012, pp. 732–745. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-97304-3_56
7. Al-Sahaf, H., Al-Sahaf, A., Xue, B., Johnston, M., Zhang, M.: Automatically evolving rotation-invariant texture image descriptors by genetic programming. *IEEE Trans. Evol. Comput.* **21**(1), 83–101 (2017)
8. Al-Sahaf, H., Song, A., Neshatian, K., Zhang, M.: Two-tier genetic programming: towards raw pixel-based image classification. *Expert Syst. Appl.* **39**(16), 12291–12301 (2012)
9. Al-Sahaf, H., Xue, B., Zhang, M.: A multitree genetic programming representation for automatically evolving texture image descriptors. In: Shi, Y., et al. (eds.) *SEAL 2017. LNCS*, vol. 10593, pp. 499–511. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68759-9_41
10. Al-Sahaf, H., Zhang, M., Al-Sahaf, A., Johnston, M.: Keypoints detection and feature extraction: a dynamic genetic programming approach for evolving rotation-invariant texture image descriptors. *IEEE Trans. Evol. Comput.* **21**(6), 825–844 (2017)
11. Borji, A., Cheng, M.M., Jiang, H., Li, J.: Salient object detection: a benchmark. *IEEE Trans. Image Process.* **24**(12), 5706–5722 (2015)
12. Borji, A., Cheng, M., Jiang, H., Li, J.: Salient object detection: a survey. *CoRR abs/1411.5878* (2014)
13. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. *Int. J. Comput. Vis.* **59**(2), 167–181 (2004)
14. Heikkilä, M., Pietikäinen, M., Schmid, C.: Description of interest regions with local binary patterns. *Pattern Recogn.* **42**(3), 425–436 (2009)
15. Hou, Q., Cheng, M.M., Hu, X., Borji, A., Tu, Z., Torr, P.: Deeply supervised salient object detection with short connections. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5300–5309. IEEE (2017)
16. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(11), 1254–1259 (1998)
17. Jiang, H., Wang, J., Yuan, Z., Wu, Y., Zheng, N., Li, S.: Salient object detection: a discriminative regional feature integration approach. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2083–2090 (2013)
18. Koza, J.R.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, vol. 1. MIT press, Cambridge (1992)
19. Lavrenko, V., Allan, J., DeGuzman, E., LaFlamme, D., Pollard, V., Thomas, S.: Relevance models for topic detection and tracking. In: *Proceedings of the Second International Conference on Human Language Technology Research*, pp. 115–121. Morgan Kaufmann Publishers Inc. (2002)
20. Lee, G., Tai, Y.W., Kim, J.: Deep saliency with encoded low level distance map and high level features. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 660–668 (2016)

21. Lensen, A., Al-Sahaf, H., Zhang, M., Xue, B.: Genetic programming for region detection, feature extraction, feature construction and classification in image data. In: Heywood, M.I., McDermott, J., Castelli, M., Costa, E., Sim, K. (eds.) EuroGP 2016. LNCS, vol. 9594, pp. 51–67. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30668-1_4
22. Lin, M., Zhang, C., Chen, Z.: Predicting salient object via multi-level features. *Neurocomputing* **205**, 301–310 (2016)
23. Liu, T., et al.: Learning to detect a salient object. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(2), 353–367 (2011)
24. Luo, Z., Mishra, A.K., Achkar, A., Eichel, J.A., Li, S., Jodoin, P.M.: Non-local deep features for salient object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. vol. 2, p. 7 (2017)
25. Perazzi, F., Krähenbühl, P., Pritch, Y., Hornung, A.: Saliency filters: contrast based filtering for salient region detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 733–740. IEEE (2012)
26. Song, H., Liu, Z., Du, H., Sun, G., Le Meur, O., Ren, T.: Depth-aware salient object detection and segmentation via multiscale discriminative saliency fusion and bootstrap learning. *IEEE Trans. Image Process.* **26**(9), 4204–4216 (2017)
27. Yang, C., Zhang, L., Lu, H., Ruan, X., Yang, M.H.: Saliency detection via graph-based manifold ranking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3166–3173. IEEE (2013)
28. Zhang, P., Wang, D., Lu, H., Wang, H., Ruan, X.: Amulet: aggregating multi-level convolutional features for salient object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 202–211 (2017)
29. Zhou, L., Yang, Z., Yuan, Q., Zhou, Z., Hu, D.: Salient region detection via integrating diffusion-based compactness and local contrast. *IEEE Trans. Image Process.* **24**(11), 3308–3320 (2015)
30. Zhu, W., Liang, S., Wei, Y., Sun, J.: Saliency optimization from robust background detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2814–2821. IEEE (2014)